

Wie viel Usability-Engineering braucht das Requirements-Engineering?

Chris Rupp

General Manager
SOPHIST GmbH
Vordere Cramergasse 13
90478 Nürnberg
chris.rupp@sophist.de

Abstract: Software Produkte sind nicht immer einfach und intuitiv benutzbar, was uns Nutzer oft verzweifeln und verärgern lässt. Dies liegt daran, dass wir Endbenutzer zu selten in den Entwicklungsprozess mit eingebunden werden, obwohl die Produkte für uns entwickelt werden. Unsere Ziele und Bedürfnisse werden oft gar nicht betrachtet oder kommen viel zu kurz. Doch was kann man aus Requirements Engineering Sicht tun, um dieser Problematik entgegenzuwirken? Im Rahmen eines Innovationsprojektes beschäftigen sich die SOPHISTen mit der Frage, wie viel Usability Engineering das Requirements Engineering braucht, um die Benutzbarkeit von Softwareprodukten zu stärken und dem User ein positives Nutzungserlebnis zu bieten. Wir suchen nach methodischen Schnittpunkten beider Disziplinen, um anschließend einige Usability-Engineering-Methoden in das Requirements Engineering zu transferieren. Weiter forschen wir an der Erstellung eines Vorgehensmodells, bei dem sich Usability-Engineering-Methoden und Requirements Engineering Methoden eingliedern. Sie bekommen über folgende auch für Requirements Engineers spannende Themen aus dem Usability Engineering etwas zu hören:

- Das Personas-Konzept nach Cooper – und wie es das klassische Stakeholder-Relationship-Management ergänzt
- Die Szenariobasierte Erstellung von Kontextszenarien mit anschließender Erhebung von Anforderungen
- Sowie ein Vorgehensmodell, das Usability-Engineering-Methoden und Requirements Engineering Methoden einordnet

Seien Sie gespannt, welche Zusatzerkenntnisse man aus einem professionellen Usability Engineering als Requirements Engineer ziehen kann.

Synergien nutzen – Personas als neue Anforderungsquelle

In der heutigen Zeit nimmt die Usability (= Benutzbarkeit) von Software eine immer größer werdende Rolle ein. Jeder Benutzer hat den Anspruch, dass Produkte einfach und intuitiv zu bedienen sind, denn nur so arbeitet er mit einem Produkt häufig und vor allem gerne. Dennoch gibt es unzählige Softwareprodukte, die eine schlechte Usability aufweisen und dadurch Benutzer oft verzweifeln lassen. Überwiegend liegt dies daran, dass die Ziele, Bedürfnisse und Verhaltensweisen der Benutzer nicht analysiert werden und diese wichtigen Ergebnisse im weiteren Entwicklungsprozess fehlen. Zudem werden die Disziplinen Requirements-Engineering (RE) und Usability-Engineering getrennt voneinander betrachtet und durchgeführt, sodass es bisher kein Vorgehen gibt, welches die Usability-Aspekte frühzeitig in die Anforderungsermittlung integriert.

Was können wir aus RE-Sicht tun, um diesem Problem entgegenzuwirken?

Wir SOPHISTen haben eine erste Version eines Vorgehensmodells entwickelt, welches Usability-Engineering-Methoden systematisch in das Requirements-Engineering integriert. Dabei sollen sich die Methoden der beiden Disziplinen bestmöglich ergänzen, sodass eine qualitativ hochwertige Anforderungsspezifikation entsteht, die als Basis für den weiteren Entwicklungsprozess zur Verfügung steht. Im ersten Teil unserer Forschungsarbeit haben wir die Ermittlungs- und Dokumentationstechniken beider Disziplinen genauer betrachtet und auf Schnittstellen und Unterschiede hin untersucht. Da sich die genannten Teilbereiche in der Analyse-Phase des allgemeinen Entwicklungsprozesses befinden, war dieses Vorgehen möglich. Bei der Untersuchung stellte sich heraus, dass die Ermittlungstechniken der beiden Disziplinen fast identisch sind. Somit kann das Requirements-Engineering auf Ergebnissen des Usability-Engineerings aufbauen und umgekehrt. Problematisch dagegen schien, dass der Fokus der Ermittlung und Dokumentation jeweils ein ganz anderer ist. Das Requirements-Engineering stellt in der Systemanalyse das System mit seinen Funktionalitäten in den Mittelpunkt und arbeitet eher top-down. Im Usability-Engineering wird dagegen der Fokus auf die Benutzer gelegt und bottom-up vorgegangen. Doch genau diese unterschiedliche Fokussierung eröffnet ein enormes Potential für die kombinierte Variante. So entstehen durch die verschiedenen Schwerpunkte neue Betrachtungswinkel, die als Basis für weitere Anforderungen dienen.



Abbildung 1: Übersicht der methodischen Schnittstellen

Abbildung 1 zeigt drei Schnittpunkte, an denen wir die beiden Disziplinen methodisch kombiniert haben. Die erste Schnittstelle „Anforderungsquellen ermitteln“ zeigt auf, wie sich das klassische Stakeholder-Relationship-Management des Requirements Engineering [Ru09], [PC10] mit dem Persona-Konzept von Alan Cooper bestmöglich ergänzen lässt und wann Personas als neue Anforderungsquellen verwendet werden können. Um im nächsten Schritt eine optimale Kontextabgrenzung durchführen zu können, kombinieren wir die Ergebnisse aus Stakeholder-Interviews und Persona-Beschreibungen. Vor allem die Ziele und das reale Arbeitsumfeld des Benutzers sind dabei von zentraler Bedeutung. Außerdem ist es zu diesem Zeitpunkt manchmal sinniger die potentiellen Anforderungsquellen klar zu priorisieren anstatt später zu viele sich widersprechende Informationen unterschiedlicher Quelle zu konsolidieren. Im letzten Schritt „Anforderungen erheben“ zeigen wir auf, welche Faktoren der Benutzeranalyse als Basis für funktionale und nichtfunktionale Anforderungen dienen und wie diese richtig extrahiert werden.

In diesem Artikel betrachten wir ausschließlich den ersten Schnittpunkt „Anforderungen ermitteln“. Dabei geben wir zuerst einen kurzen Überblick über das Stakeholder-Relationship-Management sowie über die Grundlagen des Persona-Konzeptes, bevor wir Ihnen unser Ergebnis der kombinierten Variante präsentieren.

Was sind Stakeholder und warum sind sie so wichtig?

Im klassischen RE sind Stakeholder Personen oder Organisationen, die Einfluss auf Anforderungen haben. Sie dienen neben Dokumenten und Systemen in Betrieb als wichtigste Anforderungsquelle. Da eine unvollständige Stakeholder-Analyse zu einer lückenhaften Anforderungsspezifikation führen kann, wird versucht, alle Stakeholder mit Hilfe einer Stakeholder-Checkliste zu erfassen. Die ermittelten Stakeholder werden dann in der sogenannten Stakeholder-Liste dokumentiert, die neben essentiellen Attributen, wie Funktion (Rolle) und Wissensgebiet, auch viele organisatorische Details, wie Verfügbarkeit enthält. Da Stakeholder in den gesamten Entwicklungsprozess mit einbezogen werden, sollte die Stakeholder-Liste stets aktuell gehalten werden. Außerdem entscheidet der Projektleiter anhand dieser Liste über die Freigabe von Stakeholder-Ressourcen [Ru09].

Um die Theorie nicht so abstrakt zu halten, verwenden wir folgendes Beispiel: Für eine Restaurantkette soll ein System entwickelt werden, welches die Tätigkeiten des Kellners größtenteils ersetzt (z. B. Bestellung entgegennehmen, Weinempfehlung aussprechen), indem der Gast beim Betreten des Restaurants ein Bedienteil in die Hand bekommt. Mit dem Bedienteil tätigt der Gast z. B. seine Bestellungen, informiert sich über die Herkunft der Lebensmittel und bezahlt die Rechnung. Neben den Tätigkeiten des Kellners soll das System auch administrative Tätigkeiten der weiteren Angestellten systemisch unterstützen (z. B. Lebensmittel bestellen, Urlaub planen).

Im Restaurantbeispiel sind die Stakeholder zum einen die Mitarbeiter des Restaurants (z. B. Geschäftsführer, Koch, Thekenmitarbeiter) sowie alle Gäste. Aber auch Wartungskräfte und Entwickler sind Stakeholder, da sie aufgrund ihrer Funktionen Einfluss auf Anforderungen haben. Nachdem das System verschiedenen gesetzlichen Vorschriften entsprechen muss, wird z. B. Hr. Herold als Repräsentant für das Jugendschutzgesetz mit in die Stakeholder-Liste aufgenommen. Da die Funktion des Kellners durch das System abgelöst werden soll, bleibt er in der Stakeholder-Liste als Anforderungsquelle bestehen. Alle ermittelten Stakeholder werden in der Stakeholder-Liste dokumentiert und anschließend befragt. Abbildung 2 zeigt einen Ausschnitt der Stakeholder-Liste. Bei der Befragung von Stakeholdern im klassischen Requirements-Engineering wird meist der Fokus auf Systemanforderungen und weniger auf die Bedürfnisse und Verhaltensweisen der Benutzer gelegt.

| Funktion (Rolle) | Name | Kontaktdaten | Wissensgebiet | Begründung |
|--------------------------|--------------------|-----------------|-----------------------------------------------|----------------------------------------------------|
| Geschäftsführer | Antonio Müller | 47mueller@bl.de | Kennt alle Abläufe innerhalb des Restaurants | Entscheidung über Realisierung, ist Geldgeber |
| Koch | Franco Ferro | ferro@bl.de | Experte für den Lebensmitteleinkauf | Anwender des Systems, muss Lebensmittel bestellen |
| Entwickler | Franz Huber | huber@4soft.de | Objektorientierte Programmierung | Technische Realisierung des Systems |
| Kellner | Tobias Wegerer | tobi-w@aol.de | Speisenservice und Gästebetreuung | Das System soll den Kellner ersetzen. |
| Rezeptionist | Michaela Heilmeier | 0171 5628452 | Reservierungen entgegennehmen | Anwender des Systems, für Reservierungen zuständig |
| Thekenmitarbeiter | Matthias Hansen | 0911 3457890 | Ausschank, Getränkeservice und Gästebetreuung | Anwender des Systems, für Getränke zuständig |
| Service-techniker | Thorsten Keller | tk22@gmx.de | Wartung und Reparatur technischer Geräte | Wartung und Reparatur der Endgeräte |
| Gast | Anna Schweiger | 0174 8813401 | Administrative Tätigkeiten am PC | Anwenderin des Endgeräts, will Speisen bestellen |
| Jugendschutzbeauftragter | Benjamin Herold | b.herold@js.com | Experte des Jugendschutzgesetzes | Jugendschutzgesetz muss berücksichtigt werden |

Abbildung 2: Ausschnitt einer Stakeholder-Liste


Nachdem wir Ihnen einen kurzen Einblick in das Stakeholder-Relationship-Management gegeben haben, betrachten wir nun eine Methode des Usability-Engineerings genauer. Das Usability-Engineering arbeitet mit dem User-Modell Personas von Alan Cooper [Co07], um die Bedürfnisse der Benutzer zu dokumentieren.

Was sind Personas?

Personas sind User-Archetypen, die die verschiedenen Ziele und beobachteten Verhaltensmuster der potentiellen Benutzer und Kunden beschreiben [Go09]. Bei Personas handelt es sich also nicht um reale Menschen, sondern um Idealtypen, die aber auf den Verhaltensweisen und Zielen echter Menschen basieren. So werden zu Beginn der Systementwicklung Markt- und Designforschung betrieben. Denn bevor ein Produkt den Kundenwünschen nach entwickelt werden kann, müssen erst verschiedene Daten der potenziellen und tatsächlichen Benutzer des Produkts vorliegen. So steht in der sogenannten Research-Phase (Forschungsphase) die Suche nach Bedürfnissen, Ansichten und Zielen der Benutzer im Mittelpunkt [Go09].

Auf das Restaurantsystem bezogen werden die Benutzer des Systems (Mitarbeiter und Gäste) nach ihren Aktivitäten, Zielen, technischen Einstellungen, kognitiven Fähigkeiten, ihrem Umgang mit IT-Systemen, aber auch ihren Ängsten befragt. Besonders wichtig ist es, das mentale Modell der Benutzer – ihre Vorstellung wie das Restaurantsystem funktioniert – zu erforschen. Denn liegt die technische Umsetzung des Restaurantsystems später nahe bei den Vorstellungen der Benutzer, können diese das System intuitiv bedienen. Wichtig bei der Beobachtung und Befragung von Benutzern ist es auch, dass sie in ihrer Umgebung, dem sogenannten Nutzungskontext, betrachtet werden. So wird der Koch bei seinen täglichen Arbeitsaufgaben in der Küche beobachtet und gegebenenfalls dazu befragt. Durch die Kontextbetrachtung können die Ziele der Benutzer besser verstanden werden. Die gesamten ermittelten Informationen werden zusammengefasst und anschließend pro Persona in einem sogenannten Persona-Template visualisiert. Die Abbildung 3 zeigt die Persona „Koch“.

Mit Hilfe von Personas kann über verschiedene Benutzertypen und ihre Bedürfnisse kommuniziert werden, um dann zu entscheiden, welcher für das Design von Form und Verhalten am wichtigsten ist [Go09].

| | | |
|-------------------------|-----------------------|--------------------------------------------------------------------------------------|
| Rolle | Koch |  |
| Name: | Tommaso Zanolla | |
| Demografische Variablen | | |
| Alter: | 34 | |
| Familienstand: | Verheiratet, 2 Kinder | |
| Verdienst: | Gering | |
| Wohnort: | Außerhalb der Stadt | |
| Herkunft: | Italien | |
| Verhaltensvariablen | | |

| | |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Aktivitäten: | <i>Einkauf und Lagerhaltung verwalten;</i> Koch ist dafür zuständig, dass die Woche über genug Lebensmittel und Getränke zur Verfügung stehen. Außerdem ist er auch für die Vorratshaltung verantwortlich. Dazu steht in jedem Restaurant eine Lieferantenakte zur Verfügung. |
| | <i>Gericht zubereiten;</i> Der Koch bereitet das bestellte Gericht zu. |
| | <i>Speisekarte verändern;</i> Der Koch muss sich im Abstand von 12 Wochen eine neue Spezialität aneignen. Außerdem kann er bei der Geschäftsleitung die Streichung eines Gerichts beantragen, falls dieses nicht oft bestellt wird. |
| | <i>Urlaub planen;</i> Urlaubsantrag muss drei Wochen vorher im Kalender der Stammfiliale eingetragen werden. |
| | <i>Filiale zuweisen;</i> Jeder Mitarbeiter muss sich wöchentlich in der Stammfiliale in die Planungstabelle eintragen und sich dabei einer Filiale zuweisen, in der er die Woche über arbeiten möchte. |
| | <i>Bestellungen aufgeben;</i> Bestellungen, die nicht Lebensmittel betreffen, werden auf einem vorgegebenen Formular eingetragen. |
| Einstellungen: | <i>Technischer Fortschritt;</i> Der Koch denkt positiv über den technischen Fortschritt und findet die Idee, ein zentrales System zu schaffen sehr gut. Jedoch hat er Angst, dass kurzfristige Bestellungen (Einkäufe) verloren gehen könnten, falls es technische Probleme gibt. |
| Fähigkeiten: | <i>Bildungsstand;</i> Der Koch besitzt eine abgeschlossene Berufsausbildung. |
| | <i>Fähigkeit zu lernen;</i> Im visuellen Lernen sehr gut. |
| Ziele: | <i>Zentrale Einkaufsliste;</i> Der Koch möchte zu jeder Zeit auf die „Lieferantentabelle“ restaurantübergreifend zugreifen können, um schneller arbeiten zu können. |
| | <i>Vorratshaltung anzeigen;</i> Der Koch möchte, dass der Vorratsbestand vom System angezeigt wird. |
| | <i>Alle Spezialitäten anzeigen;</i> Der Koch möchte seine bisherigen Spezialitäten angezeigt bekommen, um sich seiner Kreativität bewusst zu werden. |
| | <i>Preisvergleich beim Einkauf;</i> Der Koch möchte beim Einkauf einen Preisvergleich der Produkte angezeigt bekommen, um günstiger einkaufen zu können. |

| | |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Können: | <i>Umgang mit dem Computer;</i> Der Koch besitzt privat einen Computer und nutzt diesen zur täglichen Administration (E-Mails, Word, Excel, Fotos hochladen). Außerdem nutzt der Koch unregelmäßig soziale Netzwerke. Bei kleineren technischen Problemen braucht er keine fremde Hilfe. |
| Können: | <i>Umgang mit dem Computer;</i> Der Koch besitzt privat einen Computer und nutzt diesen zur täglichen Administration (E-Mails, Word, Excel, Fotos hochladen). Außerdem nutzt der Koch unregelmäßig soziale Netzwerke. Bei kleineren technischen Problemen braucht er keine fremde Hilfe. |
| | <i>Mobiltelefon;</i> Der Koch besitzt ein Smartphone und benutzt dieses öfters als seinen Computer, um ins Internet zu gehen. |
| Mentales Modell: | <i>Vorstellung des Systems;</i> Der Koch hat die Vorstellung, dass die Dokumente in dem System wie ein Stapel Papier aufgebaut sind. In dem Stapel Papier befinden sich alle Tabellen und Listen, die es jetzt in Papierform auch gibt. |
| Physische Umgebung (Nutzungskontext): | In der Küche läuft meistens Radio im Hintergrund. Bei verschiedenen Arbeitsschritten ist der Geräuschpegel lauter (Spülen, Braten, Geschirr einräumen, Mixen,...). Außerdem gibt es einige Geräte, die Töne von sich geben (Gefriertruhe, falls zu lange offen, Backofen, nach eingestellter Zeit,...). Der Koch hat in der Küche verschiedene Arbeitsplätze. Diese sind in Lebensmittel- und Schmutzbereich unterteilt. Im Lebensmittelbereich müssen verschiedenste Hygienemaßnahmen durchgeführt werden, um mit Lebensmitteln zu arbeiten. Flüssigkeiten, Fettspritzer und extreme Temperaturen stellen eine Gefahr für Endgeräte dar. Der Koch legt bestimmte Wege in der Küche zurück. |

Abbildung 3: Fertige Beschreibung der Persona „Koch“

Nachdem wir Ihnen das Stakeholder-Relationship-Management sowie die Modellierung von Personas vorgestellt haben, kombinieren wir nun die beiden Methoden, um eine vollständige und fundierte Benutzeranalyse zu erhalten.

Wie können sich das Stakeholder-Relationship-Management und Personas ergänzen?

Abbildung 4 zeigt unseren Vorschlag für das kombinierte Vorgehen mit entsprechenden Prozessschritten und Ergebnisartefakten.

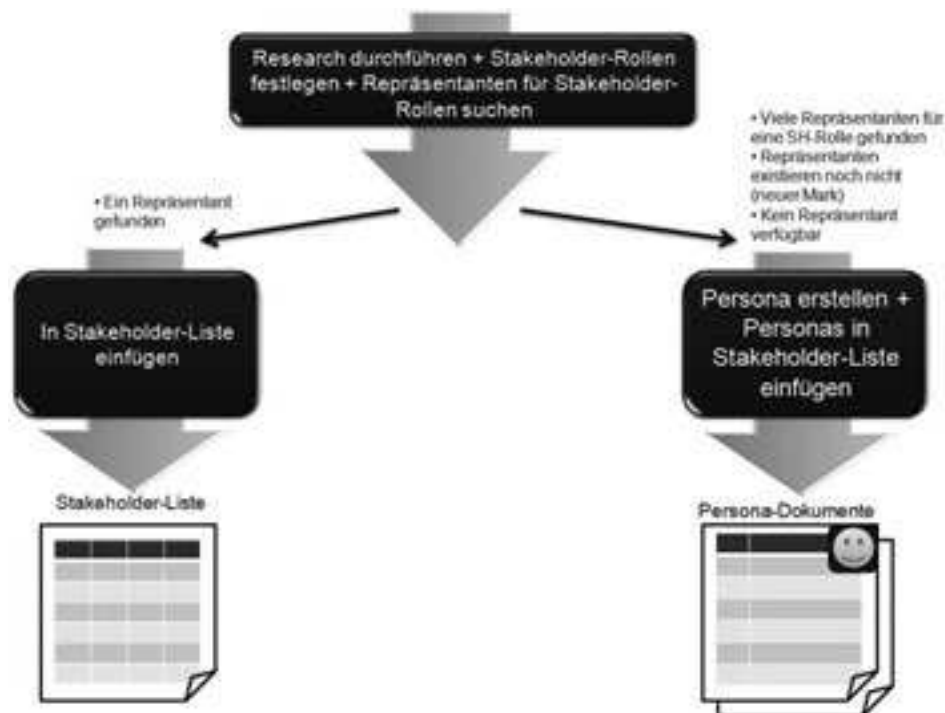


Abbildung 4: Methodische Kombination von Stakeholder-Relationship-Management und Personas

Als erstes werden alle Stakeholder-Rollen mit Hilfe der Stakeholder-Checkliste ermittelt und dazu Repräsentanten gesucht. Als weitere Informationsquelle werden hier auch die Techniken des Usability-Engineerings, wie Ethnografie oder Marktforschung eingesetzt, um eine vollständige Stakeholder-Auflistung zu erreichen. Vor allem neue Stakeholder-Rollen aus noch zu erschließenden Märkten können damit ermittelt werden. Wurden Repräsentanten für eine Stakeholder-Rolle gefunden, so werden sie in die Stakeholder-Liste eingetragen. Als Ergebnisartefakt entsteht nach diesem Schritt eine Stakeholder-Liste (vgl. Abbildung 2).

In der Stakeholder-Liste befinden sich nun z. B. zehn Köche und 30 Gäste. Existieren mehrere Stakeholder für eine Funktion (z. B. Koch) oder gibt es eine Stakeholder-Rolle (im Beispiel Gast), die viele gut voneinander abgrenzbare Untergruppen (z. B. Stammgäste oder Laufkundschaft) repräsentiert, werden dafür Personas erstellt. Hier

erzeugt eine Persona-Modellierung einen fundierteren Einblick in die Wünsche dieser wichtigen Personengruppe, als das Gespräch mit einem einzelnen Repräsentanten.

Dazu werden die Benutzer, die das Restaurantsystem regelmäßig benutzen, in der Stakeholder-Liste markiert (vgl. Abbildung 5). Das sind Geschäftsführer, Koch, Rezeptionist, Thekenmitarbeiter und Gast. Der Servicetechniker bleibt in diesem Fall außen vor, da er das System nicht so häufig benutzt. In dem Fall, dass es für einen Benutzer genau einen realen Stakeholder gibt, wird dieser nicht zur Persona, sondern wird weiterhin als Stakeholder betrachtet und in der Stakeholder-Liste geführt (z. B. Geschäftsführer). Eine Persona-Modellierung ist unter diesen Umständen überflüssig, da es einfacher ist, den einen realen Stakeholder zu befragen.

| Produkt | Stakeholder-Rollen | Begründung |
|-------------------|--------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Restaurant-system | Mitarbeiter | Mitarbeiter sind alle Angestellte des Unternehmens und werden somit als eine generalisierte Rolle betrachtet. |
| | Koch | Koch ist für die Bestellung und Lagerverwaltung der Lebensmittel zuständig. |
| | Rezeptionist | Rezeptionist hat mehr Befugnisse als die anderen Mitarbeiter und ist vor allem für die Reservierungen zuständig. |
| | Thekenmitarbeiter | Thekenmitarbeiter führt bestimmte Servicetätigkeiten aus. |
| | Gast | Gast nimmt Produkte und Dienstleistungen des Restaurants in Anspruch und wird somit als eine generalisierte Rolle betrachtet. |

Abbildung 5: Persona-Einteilung abgeleitet aus Stakeholder-Liste

Existieren keine Repräsentanten für eine Stakeholder-Funktion, da sie erst noch geprägt werden müssen (z. B. zukünftiger Markt oder zukünftige Benutzersicht), oder sind die Repräsentanten sehr schwierig zu erreichen, so werden in diesen Fällen ebenfalls Personas erstellt. Nachdem entsprechende Personas erstellt wurden, werden diese mit in die Stakeholder-Liste aufgenommen. Als Ergebnisartefakte existieren nun eine Stakeholder-Liste und mehrere Persona-Beschreibungen. Beide Dokumente behalten ihre ursprüngliche Darstellungsform bei.

Im weiteren Verlauf sollte darauf geachtet werden, dass Stakeholder und Personas gleichwertig für die Ableitung von Anforderungen betrachtet werden. Denn die Gefahr einer starken Fokussierung auf die Benutzer, die im Rahmen der Stakeholder-Analyse gefunden wurden, ist gegeben.

Abschließend können wir sagen, dass durch die Kombination von Stakeholder-Relationship-Management und Personas eine vollständige Benutzeranalyse entsteht, die als solide Basis für die weitere Produktentwicklung dient. Die beiden Disziplinen ergänzen sich dabei in ihrer Vorgehensweise perfekt, indem das Usability-Engineering an vielen Stellen immer wieder die Kreativität des Analytikers anregt und das Requirements-Engineering gleichzeitig die nötige Struktur liefert. Speziell die Benutzeranalyse liefert essentielle Daten wie Ziele, Verhaltensweisen und Bedürfnisse der Benutzer, die im klassischen Requirements-Engineering nicht so detailliert erhoben werden. Die Analyse und Modellierung von Benutzern ist somit eine wichtige Anforderungsquelle für funktionale und nichtfunktionale Anforderungen, da sie bei Entwicklungs- und Designentscheidungen hilft und die Sichtweise auf das System und seine Interaktion erweitert. Die Kombination der beiden Disziplinen erweitert nicht nur den Blickwinkel auf das System, sondern bringt Vollständigkeit für die weiteren Entwicklungsphasen und sichert die Akzeptanz der Nutzer. Außerdem fördert eine Verschmelzung der beiden Disziplinen das Miteinander der Projektteilnehmer langfristig.

Unsere Forschungsarbeit zur methodischen Kombination der Disziplinen Requirements-Engineering und Usability-Engineering ist noch nicht abgeschlossen. So arbeiten wir aktuell an Themenbereichen wie Form- und Verhaltensspezifikation sowie Usability-Tests. Bei Interesse an einer Forschungsk Kooperation zu diesem Thema oder bei Interesse an weiteren Informationen wenden Sie sich bitte per E-Mail an heureka@sophist.de.

Literaturverzeichnis

- [Ru09] Rupp, Chris / SOPHISTen, die: Requirements- Engineering und –Management – Professionelle, Iterative Anforderungsanalyse für die Praxis, Hanser, Nürnberg, 2009
- [PC10] Pohl, Klaus / Rupp, Chris: Basiswissen Requirements Engineering – Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt.verlag GmbH, Heidelberg, 2010
- [Co07] Cooper, Alan / Reimann, Robert / Cronin David: About Face – Interface und Interaction Design, mitp, Heidelberg, 2007
- [Go09] Goodwin, Kim: Designing For The Digital Age, Wiley Publishing, Inc., Indianapolis, 2009

